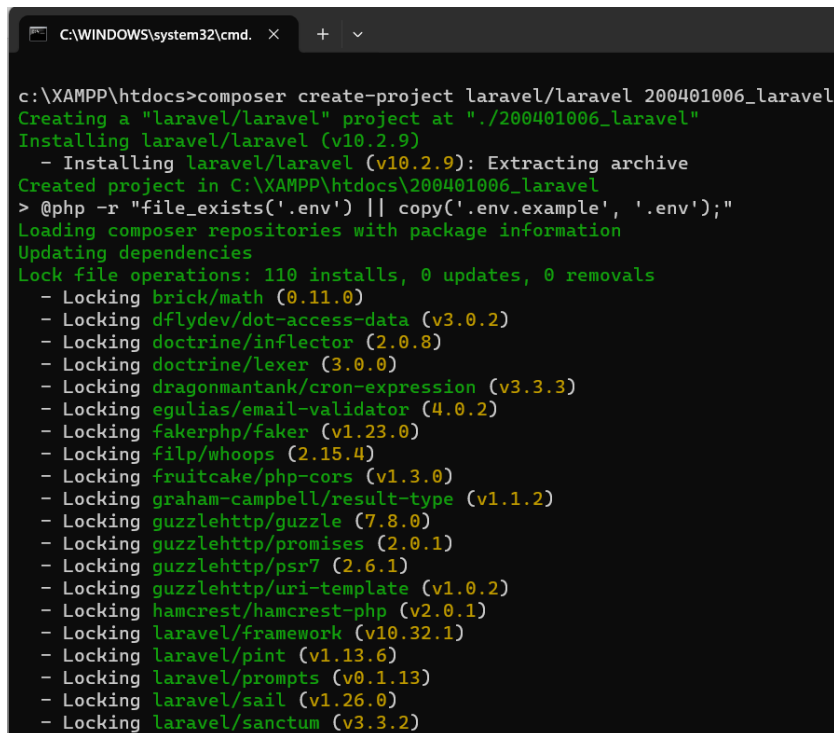


Nama : Muhammad Rizky Firmansyah

NIM : 200401006

1. Install Laravel dengan prefix nama aplikasi sesuai NIM, contoh200401122_laravel

Jawab :

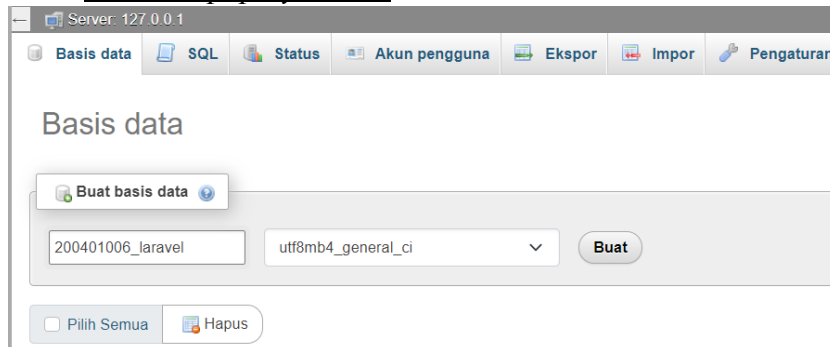


```
c:\XAMPP\htdocs>composer create-project laravel/laravel 200401006_laravel
Creating a "laravel/laravel" project at "C:\XAMPP\htdocs\200401006_laravel"
Installing laravel/laravel (v10.2.9)
- Installing laravel/laravel (v10.2.9): Extracting archive
Created project in C:\XAMPP\htdocs\200401006_laravel
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 110 installs, 0 updates, 0 removals
- Locking brick/math (0.11.0)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/inflector (2.0.8)
- Locking doctrine/lexer (3.0.0)
- Locking dragonmantank/cron-expression (v3.3.3)
- Locking egulias/email-validator (4.0.2)
- Locking fakerphp/faker (v1.23.0)
- Locking filp/whoops (2.15.4)
- Locking fruitcake/php-cors (v1.3.0)
- Locking graham-campbell/result-type (v1.1.2)
- Locking guzzlehttp/guzzle (7.8.0)
- Locking guzzlehttp/promises (2.0.1)
- Locking guzzlehttp/psr7 (2.6.1)
- Locking guzzlehttp/uri-template (v1.0.2)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v10.32.1)
- Locking laravel/pint (v1.13.6)
- Locking laravel/prompts (v0.1.13)
- Locking laravel/sail (v1.26.0)
- Locking laravel/sanctum (v3.3.2)
```

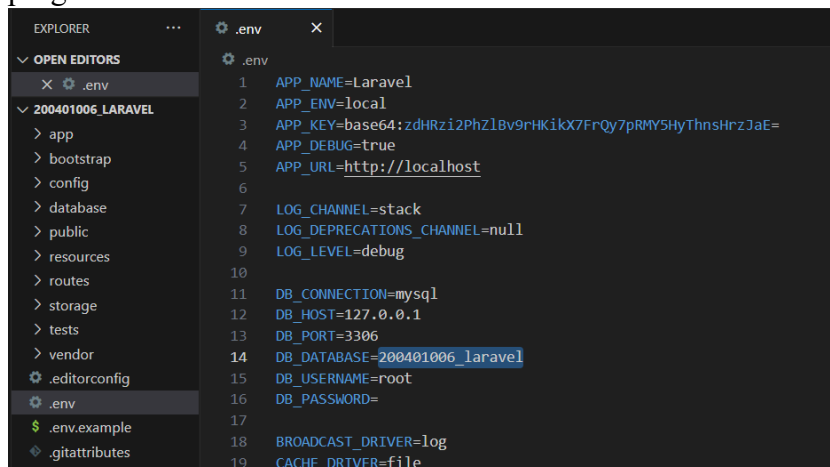
2. Buatlah database untuk aplikasi laravel dengan prefix sesuai NIM, contoh200401122_laravel

Jawab :

- Aktifkan aplikasi XAMPP
- Buka localhost/phpMyAdmin dan buat data base baru



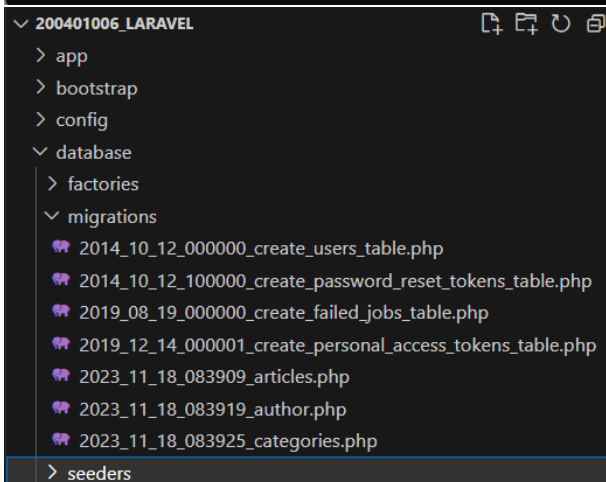
- Lalu masukan nama data base di file .env untuk menghubungkan data base dengan folder program



3. Buatlah file migration untuk table dari aplikasi tersebut dengan 3 buah table yaitu: article, authors dan categories

Jawab :

```
c:\XAMPP\htdocs>cd c:/xampp/htdocs/200401006_laravel
c:\XAMPP\htdocs\200401006_laravel>php artisan make:migration articles
INFO Migration [C:\XAMPP\htdocs\200401006_laravel\database\migrations\2023_11_18_083909_articles.php] created successfully.
c:\XAMPP\htdocs\200401006_laravel>php artisan make:migration author
INFO Migration [C:\XAMPP\htdocs\200401006_laravel\database\migrations\2023_11_18_083919_author.php] created successfully.
c:\XAMPP\htdocs\200401006_laravel>php artisan make:migration categories
INFO Migration [C:\XAMPP\htdocs\200401006_laravel\database\migrations\2023_11_18_083925_categories.php] created successfully.
c:\XAMPP\htdocs\200401006_laravel>
```



4. Buatlah fungsi Create, Read, Update, Delete (CRUD) dengan konsep MVC di Laravel Fungsi CRUD untuk article dan categories.

Jawab :

- Buat Model dan controller untuk Article & categories

```
c:\XAMPP\htdocs\200401006_laravel>php artisan make:model article
INFO Model [C:\XAMPP\htdocs\200401006_laravel\app\Models\article.php] created successfully.
c:\XAMPP\htdocs\200401006_laravel>php artisan make:model categories
INFO Model [C:\XAMPP\htdocs\200401006_laravel\app\Models\categories.php] created successfully.
c:\XAMPP\htdocs\200401006_laravel>php artisan make:controller articlecontroller
INFO Controller [C:\XAMPP\htdocs\200401006_laravel\app\Http\Controllers\articlecontroller.php] created successfully.
c:\XAMPP\htdocs\200401006_laravel>php artisan make:controller categoriescontroller
INFO Controller [C:\XAMPP\htdocs\200401006_laravel\app\Http\Controllers\categoriescontroller.php] created successfully.
```

- Lalu masukan Routing untuk menghubungkan Article & categories dengan masing masing controller nya

```
.env web.php x articlecontroller.php categoriesc
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  /*
6  |-----
7  | Web Routes
8  |-----
9  |
10 | Here is where you can register web routes for your appli
11 | routes are loaded by the RouteServiceProvider and all of
12 | be assigned to the "web" middleware group. Make somethin
13 |
14 */
15 Route::resource('articles', ArticleController::class);
16 Route::resource('categories', CategoryController::class);
17
18 Route::get('/', function () {
19     return view('welcome');
20 });
21
```

- Masukkan codingan fungsi CRUD di Controller Article & Categories
 - a. Controller

```
<?php

namespace App\Http\Controllers;

use App\Models\article;
use Illuminate\Http\Request;

class articlecontroller extends Controller
{
    public function index()
    {
        $article = article::all();
        return view('article.index', compact('article'));
    }

    public function create()
    {
        return view('article.create');
    }

    public function store(Request $request)
    {
        // Validasi data dari $request

        article::create($request->all());

        return redirect()->route('article.index')->with('success', 'article
created successfully');
    }

    public function edit(article $article)
```

```

{
    return view('article.edit', compact('article'));
}

public function update(Request $request, article $article)
{
    // Validasi data dari $request

    $article->update($request->all());

    return redirect()->route('article.index')->with('success', 'article
updated successfully');
}

public function destroy(article $article)
{
    $article->delete();

    return redirect()->route('article.index')->with('success', 'article
deleted successfully');
}
}

```

b. Categories

```

<?php

namespace App\Http\Controllers;

use App\Models\categories;
use Illuminate\Http\Request;

class categoriescontroller extends Controller
{
    public function index()
    {
        $categories = categories::all();
        return view('categories.index', compact('categories'));
    }

    public function create()
    {
        return view('categories.create');
    }

    public function store(Request $request)
    {
        // Validasi data dari $request

        categories::create($request->all());

        return redirect()->route('categories.index')->with('success', 'categories
created successfully');
    }
}

```

```
public function edit(categories $categories)
{
    return view('categories.edit', compact('categories'));
}

public function update(Request $request, categories $categories)
{
    // Validasi data dari $request

    $categories->update($request->all());

    return redirect()->route('categories.index')->with('success', 'categories
updated successfully');
}

public function destroy(categories $categories)
{
    $categories->delete();

    return redirect()->route('categories.index')->with('success', 'categories
deleted successfully');
}
}
```